



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

*nh*

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/672,183	09/25/2003	Eduard K. de Jong	SUN-040024	9848

24209 7590 02/05/2007  
GUNNISON MCKAY & HODGSON, LLP  
1900 GARDEN ROAD  
SUITE 220  
MONTEREY, CA 93940

EXAMINER
----------

CHEN, QING

ART UNIT	PAPER NUMBER
----------	--------------

2191

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	02/05/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

<b>Office Action Summary</b>	<b>Application No.</b> 10/672,183	<b>Applicant(s)</b> DE JONG, EDUARD K.	
	<b>Examiner</b> Qing Chen	<b>Art Unit</b> 2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

### Status

- 1) ☒ Responsive to communication(s) filed on 25 September 2003.
- 2a) ☐ This action is **FINAL**.                      2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

### Disposition of Claims

- 4) ☒ Claim(s) 1-48 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-48 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

### Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 25 September 2003 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All    b) ☐ Some \*    c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

### Attachment(s)

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)  | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date <u>20031215</u> . | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

1. This is the initial Office action based on the application filed on September 25, 2003.
2. **Claims 1-48** are pending.

### *Drawings*

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they do not include the following reference sign(s) mentioned in the description:

- Reference number "495" on pages 26 and 27, paragraph [0043].

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application.

4. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description:

- Reference numbers "102," "115," and "140" in Figure 1.
- Reference numbers "325," "330," "335," "340," and "370" in Figure 3.
- Reference numbers "430," "435," "440," and "470" in Figure 4.
- Reference number "550" in Figure 5C.
- Reference numbers "3110" and "3114" in Figure 31.
- Reference numbers "3200," "3220," and "3250" in Figure 32.
- Reference numbers "3610" and "3615" in Figure 36.

Art Unit: 2191

Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application.

The drawings are objected to as failing to comply with 37 CFR 1.84(p)(4) because:

- The word "Provider" is missing the letter "o" in Figure 3, Element 315.

Corrected drawing sheets in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application.

Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. Each drawing sheet submitted after the filing date of an application must be labeled in the top margin as either "Replacement Sheet" or "New Sheet" pursuant to 37 CFR 1.121(d). If the changes are not accepted by the Examiner, the Applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

*Specification*

5. The disclosure is objected to because of the following informalities:
- The specification contains the following typographical errors:
    - The attorney docket numbers for the U.S. patent applications incorporated by reference should be deleted in paragraphs [0001], [0002], [0003], [0004], and [0005].
    - The application serial numbers for the U.S. patent applications incorporated by reference are missing in paragraphs [0001], [0002], [0003], [0004], and [0005].
    - The reference number “300” should be changed to “305” on page 24, paragraph [0036], since reference number “305” is used to designate “user device” in Figure 3.
    - The element “first portion” should be changed to “second portion” on page 26, paragraph [0040], since reference number “484” is used to designate “second portion” in Figure 4.
    - The reference number “500” should be changed to “510” on page 30, paragraph [0051], since reference number “510” is used to designate “obfuscated package” in Figure 5B.
    - The word “advancing” is misspelled on page 46, paragraph [0085].
    - The reference number “2155” should be changed to “2115” on page 46, paragraph [0086], since reference number “2115” is used to designate “instruction memory stream” in Figure 21B.
    - The reference number “2515” should be changed to “2525” on page 50, paragraph [0094], since reference number “2525” is used to designate “001” in Figure 25.

Art Unit: 2191

- The reference number “2735” should be changed to “2725” on page 51, paragraph [0096], since reference number “2725” is used to designate “011” in Figure 27.
- The reference number “2915” should be changed to “2925” on page 53, paragraph [0098], since reference number “2925” is used to designate “001” in Figure 29.
- The reference number “2105” should be changed to “2305” on page 54, paragraph [0101], since reference number “2305” is used to designate “instruction location permutation table” in Figure 23.
- The reference number “3825” should be changed to “3820” on page 65, paragraph [0125] in Figure 38.

Appropriate correction is required.

6. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

#### *Claim Objections*

7. **Claims 3-6, 14-17, 25-28, 36-39, 45, and 48** are objected to because of the following informalities:

- **Claims 3, 14, 25, and 36** recite the limitation “said operation.” Applicant is advised to change this limitation to read “said modulo-n arithmetic operation” for the purpose of providing it with proper explicit antecedent basis.

- **Claims 4, 15, 26, and 37** recite the limitation “said application program.” Applicant is advised to change this limitation to read “said obfuscated application program” for the purpose of providing it with proper explicit antecedent basis.
- **Claims 5 and 6** depend on Claim 4 and, therefore, suffer the same deficiency as Claim 4.
- **Claims 16 and 17** depend on Claim 15 and, therefore, suffer the same deficiency as Claim 15.
- **Claims 27 and 28** depend on Claim 26 and, therefore, suffer the same deficiency as Claim 26.
- **Claims 38 and 39** depend on Claim 37 and, therefore, suffer the same deficiency as Claim 37.
- **Claims 45 and 48** contain a typographical error: the phrase “information used by said application program execute an obfuscated application program” should presumably read “information used by said application program to execute an obfuscated application program.”

Appropriate correction is required.

***Claim Rejections - 35 USC § 112***

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. **Claims 1-48** are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

**Claims 1, 4, 7-9, 12, 15, 18-20, 23, 26, 29-31, 34, 37, 40-42, 45, and 48** recite the limitation "at least in part." The term "at least in part" is a relative term, which renders the claims indefinite. The term "at least in part" is not defined by the claims nor does the specification provide a standard for ascertaining the requisite degree and one of ordinary skill in the art would not be able to reasonably determine the scope of the invention. In the interest of compact prosecution, the Examiner subsequently does not give any patentable weight to this limitation for the purpose of further examination.

**Claims 2 and 3** depend on Claim 1 and, therefore, suffer the same deficiency as Claim 1.

**Claims 5 and 6** depend on Claim 4 and, therefore, suffer the same deficiency as Claim 4.

**Claims 10 and 11** depend on Claim 8 and, therefore, suffer the same deficiency as Claim

8.

**Claims 13 and 14** depend on Claim 12 and, therefore, suffer the same deficiency as Claim 12.



**Claims 16 and 17** depend on Claim 15 and, therefore, suffer the same deficiency as Claim 15.

**Claims 21 and 22** depend on Claim 19 and, therefore, suffer the same deficiency as Claim 19.

**Claims 24 and 25** depend on Claim 23 and, therefore, suffer the same deficiency as Claim 23.

**Claims 27 and 28** depend on Claim 26 and, therefore, suffer the same deficiency as Claim 26.

**Claims 32 and 33** depend on Claim 30 and, therefore, suffer the same deficiency as Claim 30.

**Claims 35 and 36** depend on Claim 34 and, therefore, suffer the same deficiency as Claim 34.

**Claims 38 and 39** depend on Claim 37 and, therefore, suffer the same deficiency as Claim 37.

**Claims 43 and 44** depend on Claim 41 and, therefore, suffer the same deficiency as Claim 41.

**Claims 46 and 47** depend on Claim 45 and, therefore, suffer the same deficiency as Claim 45.

**Claims 2, 13, 24, and 35** recite the limitation “said receiving.” There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the

Examiner subsequently interprets this limitation as reading “said receiving an application program instruction” for the purpose of further examination.

**Claims 4, 15, 26, and 37** recite the limitation “the largest method.” The term “largest” is a relative term, which renders the claims indefinite. The term “largest” is not defined by the claims nor does the specification provide a standard for ascertaining the requisite degree and one of ordinary skill in the art would not be able to reasonably determine the scope of the invention. In the interest of compact prosecution, the Examiner subsequently does not give any patentable weight to this limitation for the purpose of further examination.

**Claims 5 and 6** depend on Claim 4 and, therefore, suffer the same deficiency as Claim 4.

**Claims 16 and 17** depend on Claim 15 and, therefore, suffer the same deficiency as Claim 15.

**Claims 27 and 28** depend on Claim 26 and, therefore, suffer the same deficiency as Claim 26.

**Claims 38 and 39** depend on Claim 37 and, therefore, suffer the same deficiency as Claim 37.

**Claims 8, 9, 11, 19, 20, 22, 30, 31, 33, 41, 42, and 44** recite the limitation “said application program code.” There is insufficient antecedent basis for this limitation in the claim. In the interest of compact prosecution, the Examiner subsequently interprets this limitation as reading “said transformed application program code” for the purpose of further examination.

**Claim 10** depends on Claim 8 and, therefore, suffers the same deficiency as Claim 8.

**Claim 21** depends on Claim 19 and, therefore, suffers the same deficiency as Claim 19.

**Claim 32** depends on Claim 30 and, therefore, suffers the same deficiency as Claim 30.

**Claim 43** depends on Claim 41 and, therefore, suffers the same deficiency as Claim 41.

***Claim Rejections - 35 USC § 101***

10. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

11. **Claims 23-33 and 41-48** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

**Claims 23-33** contain “means-plus-function” limitations and appear to be systems. However, it is noted that the specification does not disclose any specific corresponding structure or equivalents thereof. The recited means appear to lack the necessary physical components (hardware) to constitute a machine or manufacture under § 101. Therefore, these claim limitations can be reasonably interpreted as computer program modules—software *per se*. The claims are directed to systems of functional descriptive material *per se*, and hence non-statutory.

The claims constitute computer programs representing computer listings *per se*. Such descriptions or expressions of the programs are not physical “things.” They are neither computer components nor statutory processes, as they are not “acts” being performed. Such claimed

computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

**Claims 41-44** are directed to apparatus. However, the recited components of the apparatus appear to lack the necessary physical components (hardware) to constitute a machine or manufacture under § 101. The specification does not disclose any hardware components associated with the claimed element of an application program provider. Therefore, these claim limitations can be reasonably interpreted as computer program modules—software *per se*. Furthermore, the specification discloses that the components, process steps, and/or data structures may be implemented using firmware, computer application programs, and/or computer languages (*see Page 18, Paragraph [0015]*). Therefore, the claims are directed to apparatus of functional descriptive material *per se*, and hence non-statutory.

The claims constitute computer programs representing computer listings *per se*. Such descriptions or expressions of the programs are not physical “things.” They are neither computer components nor statutory processes, as they are not “acts” being performed. Such claimed computer programs do not define any structural and functional interrelationships between the computer program and other claimed elements of a computer, which permit the computer

Art Unit: 2191

program's functionality to be realized. In contrast, a claimed computer-readable medium encoded with a computer program is a computer element, which defines structural and functional interrelationships between the computer program and the rest of the computer, that permits the computer program's functionality to be realized, and is thus statutory. See *Lowry*, 32 F.3d at 1583-84, 32 USPQ2d at 1035.

The result of **Claims 45-48** is directed to the act of "determining the location of instruction implementation methods," which does not appear to be a tangible result so as to constitute a practical application of the idea. The act of "determining" is merely a thought or an abstract idea and does not appear to produce a tangible result even if the step of determination does occur, since the result of that determination is not conveyed in the real world. The result is a determination, which is neither used in a disclosed practical application nor made available for use in a disclosed practical application. It also does not appear that the usefulness of the determination can be realized from the claimed steps to support a disclosed specific, substantial, and credible utility so as to produce a useful result.

Therefore, the claims do not meet the statutory requirement of 35 U.S.C. § 101, since the claims are not directed to a practical application of the § 101 judicial exception producing a result tied to the physical world.

***Claim Rejections - 35 USC § 102***

12. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

13. Claims 1, 2, 4-6, 8, 9, 11-13, 15-17, 19, 20, 22-24, 26-28, 30, 31, 33-35, 37-39, 41, 42, and 44-48 are rejected under 35 U.S.C. 102(b) as being anticipated by Granger et al. (US 6,334,189).

As per Claim 1, Granger et al. disclose:

- receiving an obfuscated application program, said obfuscated application program comprising at least one instruction opcode value encoded using one of a plurality of instruction set opcode value encoding schemes (*see Column 13: 15-21, "It is therefore preferable to implement each of these components at least in-part either in pseudocode or in obfuscated machine code. In general, only one of these two techniques (pseudocode or obfuscation) will be used to hide the details of a given software function, although both techniques may be (and preferably are) used within the same application."* and 28-31, "... the details of the pseudocode instruction set (including the opcodes and instruction formats) are maintained in secrecy by the software developer." and 52-59, "... the pseudocode for performing a given function (or possibly multiple functions) is preferably stored as an encrypted pseudocode ("ECODE") data block 56 within a data table 58 of the executable application file 60. Because the pseudocode is

Art Unit: 2191

*stored within a data table, the pseudocode appears to the pirate simply as part of the application's data, and does not impair the operation of the pirate's disassembler or other analysis tool."");*

- *receiving an application program instruction corresponding to a current instruction counter value (see Column 13: 66-67 through Column 14: 1-7, "... the ECODE data block 56 preferably includes a header 66, any data 68 that is needed by the pseudocode, and the pseudocode instructions, all of which are stored in encrypted, binary form. The header includes information that is used by the interpreter 63 to process the ECODE data block 56, including an initial program counter setting of the emulated CPU, information about the arguments to be passed, and key information for decrypting the instructions 70."");*

- *selecting an instruction dispatch table based at least in part on said current instruction counter value (see Column 17: 22-30, "As the sequence of tokens for a given line is read, the token reader matches the opcode to the corresponding instruction in the internal data structure to determine the instruction format and sign information. The token reader then parses the tokens, and maps the tokens (using the mapping macros) into the 32-bit pseudocode instruction. The pseudocode instruction is then written (in unencrypted form) to an instruction list which eventually becomes part of the ECODE data block.""); and*

- *executing said application program instruction using said selected instruction dispatch table (see Column 18: 52-53, "... the SPEC executes the instruction and updates the PC (block 144).").*

As per **Claim 2**, the rejection of **Claim 1** is incorporated; and Granger et al. further disclose:

- determining whether there is another application program instruction to be executed (*see Column 18: 63-65, "For all instructions other than branch and jump instructions, the PC is incremented by one to point to the next line of the ECODE data block."*);
- advancing said current instruction counter if there is another application program instruction to be executed (*see Column 18: 63-65, "For all instructions other than branch and jump instructions, the PC is incremented by one to point to the next line of the ECODE data block."*); and
- repeating said receiving an application program instruction, said selecting and said executing after said advancing (*see Column 18: 35-39, "Once the header has been decoded, the SPEC loads the registers (block 132) with any arguments and loads the PC with the line number of the first instruction to be fetched and executed. The SPEC then enters into a main fetch/execution loop (blocks 136-144)."*).

As per **Claim 4**, the rejection of **Claim 1** is incorporated; and Granger et al. further disclose:

- wherein the number of instruction dispatch tables is based at least in part on the number of instructions in the largest method of said obfuscated application program (*see Column 16: 39-48, "... the EASM operates generally by reading one line of the text file (block 100), parsing the line (block 102), processing the parsed line to add data to a set of lists (header, data and code) that eventually become the ECODE data block (blocks 104-112), and then reading the*



Art Unit: 2191

*next line of the file (block 100). After all of the lines of the text file have been processed, the EASM merges and encrypts the header, data and code lists (blocks 118 and 120) to generate the ECODE data block 56.").*

As per **Claim 5**, the rejection of **Claim 4** is incorporated; and Granger et al. further disclose:

- wherein said number of instruction dispatch tables is greater than or equal to said number of instructions (*see Column 16: 39-48, "... the EASM operates generally by reading one line of the text file (block 100), parsing the line (block 102), processing the parsed line to add data to a set of lists (header, data and code) that eventually become the ECODE data block (blocks 104-112), and then reading the next line of the file (block 100). After all of the lines of the text file have been processed, the EASM merges and encrypts the header, data and code lists (blocks 118 and 120) to generate the ECODE data block 56.").*

As per **Claim 6**, the rejection of **Claim 5** is incorporated; and Granger et al. further disclose:

- wherein said number of instruction dispatch tables equals said number of instructions (*see Column 16: 39-48, "... the EASM operates generally by reading one line of the text file (block 100), parsing the line (block 102), processing the parsed line to add data to a set of lists (header, data and code) that eventually become the ECODE data block (blocks 104-112), and then reading the next line of the file (block 100). After all of the lines of the text file have been*

*processed, the EASM merges and encrypts the header, data and code lists (blocks 118 and 120) to generate the ECODE data block 56."*

As per **Claim 8**, Granger et al. disclose:

- reading an application program comprising code (*see Column 19: 65-67, "The C source code file 158 is then processed using a special de-optimizing cross-compiler 160 (the obfuscation tool) to generate an obfuscated C source code file 162."*);
- determining a plurality of dispatch tables associated with said application program (*see Column 17: 22-30, "As the sequence of tokens for a given line is read, the token reader matches the opcode to the corresponding instruction in the internal data structure to determine the instruction format and sign information. The token reader then parses the tokens, and maps the tokens (using the mapping macros) into the 32-bit pseudocode instruction. The pseudocode instruction is then written (in unencrypted form) to an instruction list which eventually becomes part of the ECODE data block."*);
- transforming said application program into application program code configured to utilize said plurality of dispatch tables during application program execution to determine the location of instruction implementation methods to be executed based at least in part on a current instruction counter value (*see Column 20: 5-9, "... the obfuscation tool 160 may, for example, output obfuscated machine-level code (which may optionally be in a pseudocode language), or may output obfuscated code in a different high-level language."*); and

- sending said transformed application program code (*see Column 24: 29-31, "... the workstations 202 may, for example, be in the form of network computers that download copies of the application just prior to execution."*).

As per **Claim 9**, the rejection of **Claim 8** is incorporated; and Granger et al. further disclose:

- wherein said determining further comprises determining the encoding of said plurality of dispatch tables based at least in part on a relative frequency of instructions in said transformed application program code (*see Column 17: 8-12, "The internal data structure defines the instruction set of the EASM, and specifies which of the five instruction formats is to be used to encode the assembly language instruction into a 32-bit pseudocode instruction."*).

As per **Claim 11**, the rejection of **Claim 8** is incorporated; and Granger et al. further disclose:

- after said transforming, applying a cryptographic process to said transformed application program code together with a cryptographic key to create an encrypted obfuscated application program (*see Column 24: 32-35, "The server 200 runs a license management server program 208 ("LM server") that dispatches encrypted authorization certificates to the workstations 202 in response to requests generated by the application 206."*); and
- said sending comprises sending said encrypted obfuscated application program (*see Column 25: 10-12, "Upon receiving the authorization certificate at the workstation 202, the LM client 210 decrypts and decodes the certificate to ensure that the certificate is valid."*).

**Claims 12, 13, and 15-17** are program storage device claims corresponding to the method claims above (Claims 1, 2, and 4-6) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1, 2, and 4-6.

**Claims 19, 20, and 22** are program storage device claims corresponding to the method claims above (Claims 8, 9, and 11) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 8, 9, and 11.

**Claims 23, 24, and 26-28** are apparatus claims corresponding to the method claims above (Claims 1, 2, and 4-6) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1, 2, and 4-6.

**Claims 30, 31, and 33** are apparatus claims corresponding to the method claims above (Claims 8, 9, and 11) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 8, 9, and 11.

**Claims 34, 35, and 37-39** are apparatus claims corresponding to the method claims above (Claims 1, 2, and 4-6) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 1, 2, and 4-6.

**Claims 41, 42, and 44** are apparatus claims corresponding to the method claims above (Claims 8, 9, and 11) and, therefore, are rejected for the same reasons set forth in the rejections of Claims 8, 9, and 11.

As per **Claim 45**, Granger et al. disclose:

- a data structure stored in said memory, said data structure including information used by said application program to execute an obfuscated application program, said data structure comprising application program code configured to utilize a plurality of dispatch tables during execution of said obfuscated application program to determine the location of instruction implementation methods to be executed based at least in part on a current instruction counter value (*see Column 17: 8-12, "The internal data structure defines the instruction set of the EASM, and specifies which of the five instruction formats is to be used to encode the assembly language instruction into a 32-bit pseudocode instruction." and 20-21, "A similar data structure is used by the SPEC to process the instructions."; Column 18: 22-24, "In operation, the SPEC decrypts and processes an ECODE data block (generated by the EASM) that is stored in the computer's local memory."*).

As per **Claim 46**, the rejection of **Claim 45** is incorporated; and Granger et al. further disclose:

- wherein said data structure further comprises a cryptographic key and protected data, said protected data encrypted using said cryptographic key (*see Column 17: 44-56, "... the SPEC decrypts and executes the instructions line-by-line. It is therefore desirable to use a relatively simple encryption algorithm to encrypt the instructions and data, since the use of a more complex algorithm would reduce instruction throughput. The algorithm used for this purpose is preferably a simple XOR algorithm that uses a key value stored in the header 66 and the position of the data/instruction line in the ECODE list. As indicated above, the key value is generated automatically by the EASM."*).

As per **Claim 47**, the rejection of **Claim 45** is incorporated; and Granger et al. further disclose:

- wherein said data structure further comprises an obfuscation descriptor that indicates an obfuscation method used to create said obfuscated application program (*see Column 15: 15-19, "... these tools (and the interpreter 62) can be written such that software developers can freely modify the details (opcodes, instruction formats, encryption methods, etc.) of the pseudocode so that the tools can be made publicly available."*).

As per **Claim 48**, Granger et al. disclose:

- a data structure stored in said memory, said data structure including information used by said application program to execute an obfuscated application program, said data structure comprising a plurality of dispatch tables used during execution of said obfuscated application program to determine the location of instruction implementation methods to be executed based at least in part on a current instruction counter value (*see Column 17: 8-12, "The internal data structure defines the instruction set of the EASM, and specifies which of the five instruction formats is to be used to encode the assembly language instruction into a 32-bit pseudocode instruction."* and 20-21, "A similar data structure is used by the SPEC to process the instructions."; Column 18: 22-24, "In operation, the SPEC decrypts and processes an ECODE data block (generated by the EASM) that is stored in the computer's local memory.").

***Claim Rejections - 35 USC § 103***

14. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

15. **Claims 3, 14, 25, and 36** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Granger et al.** (US 6,334,189) in view of **Folmsbee** (US 6,308,256).

As per **Claim 3**, the rejection of **Claim 1** is incorporated; and **Granger et al.** further disclose:

- selecting the instruction dispatch table associated with the result of said modulo-n arithmetic operation (*see Column 17: 33-40, "Once the last line of the text file has been processed, the EASM generates and writes the X-REF and DIS files (block 116), and concatenates the header, data and instruction lists to form a single ECODE list (not shown)."*).

However, **Granger et al.** do not disclose:

- performing modulo-n arithmetic operation on said current instruction counter value, where n is the number of dispatch tables, each of said dispatch tables associated with a unique number between 0 and n-1.

**Folmsbee** discloses:

- performing modulo-n arithmetic operation on said current instruction counter value, where n is the number of dispatch tables, each of said dispatch tables associated with a unique

Art Unit: 2191

number between 0 and  $n-1$  (see Column 8: 12-13, "... the program counter may increment by amounts from 2 to 18 (modulo  $9*4$ ).").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Folmsbee into the teaching of Granger et al. to include performing modulo- $n$  arithmetic operation on said current instruction counter value, where  $n$  is the number of dispatch tables, each of said dispatch tables associated with a unique number between 0 and  $n-1$ . The modification would be obvious because one of ordinary skill in the art would be motivated to load address locations for instructions into memory, which conform to the program counter incrementation plan (see Folmsbee – Column 8: 5-9).

**Claims 14, 25, and 36** are rejected for the same reason set forth in the rejection of Claim 3.

16. **Claims 7, 18, 29, and 40** are rejected under 35 U.S.C. 103(a) as being unpatentable over Granger et al. (US 6,334,189).

As per **Claim 7**, the rejection of **Claim 1** is incorporated; however, Granger et al. do not disclose:

- wherein the number of instruction dispatch tables is based at least in part on an amount of available memory.

Official Notice is taken that it is old and well known within the computing art to correlate the size of data with the amount of available memory. Memory components are constantly being



Art Unit: 2191

monitored by a computer system, so that appropriate actions can be taken in the event that portions of the memory become unavailable. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the number of instruction dispatch tables is based at least in part on an amount of available memory. The modification would be obvious because one of ordinary skill in the art would be motivated to prevent data loss.

**Claims 18, 29, and 40** are rejected for the same reason set forth in the rejection of Claim 7.

17. **Claims 10, 21, 32, and 43** rejected under 35 U.S.C. 103(a) as being unpatentable over **Granger et al.** (US 6,334,189) in view of **Chen** (US 5,913,064).

As per **Claim 10**, the rejection of **Claim 8** is incorporated; however, **Granger et al.** do not disclose:

- wherein said determining further comprises filtering said plurality of dispatch tables to flatten the frequency distribution of instructions over said transformed application program code.

**Chen** discloses:

- wherein said determining further comprises filtering said plurality of dispatch tables to flatten the frequency distribution of instructions over said transformed application program code (*see Column 5: 61-67, "A function determines whether filtering is performed depending*

Art Unit: 2191

*upon the number of operands specified either in the source operand stack or source object fields. The greater the number of source operands, the less likely the instruction will be "filtered". At step 34, if the instruction is "filtered" out in this manner, the instruction is not stored, and a new instruction is generated at step 26. ").*

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the teaching of Chen into the teaching of Granger et al. to include wherein said determining further comprises filtering said plurality of dispatch tables to flatten the frequency distribution of instructions over said transformed application program code. The modification would be obvious because one of ordinary skill in the art would be motivated to minimize generation of instructions, which will most easily pass typechecking, since these instructions will be "over-generated" (see Chen – Column 5: 55-59).

**Claims 21, 32, and 43** are rejected for the same reason set forth in the rejection of Claim 10.

### ***Conclusion***

18. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

A. **Drake (US 6,006,328)** discloses a program, and the system and method for creating the program, having increased security features to prevent ID data (as defined) eavesdropping and/or theft and/or ensure authenticity.

B. **Diersch et al.** (US 6,101,606) disclose a system for securing protected software from unauthorized, i.e. unlicensed, use in computer networks, e.g. networks of UNIX workstations.

C. **Nardone et al.** (US 6,175,925) disclose a tamper resistant player for scrambled contents.

D. **Sigbjørnsen et al.** (US 6,266,416) disclose an arrangement to protect software, particularly freely distributed application software, against utilization without permission of the copyright holder.

E. **Chow et al.** (US 6,594,761) disclose a method and system of making computer software resistant to tampering and reverse-engineering.

F. **Collberg et al.** (US 6,668,325) disclose methods and apparatus for increasing the structural and logical complexity of software by inserting, removing, or rearranging identifiable structure or information from the software in such a way as to exacerbate the difficulty of the process of decompilation or reverse engineering.

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Wei Zhen, can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2191

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the TC 2100 Group receptionist whose telephone number is 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).



WEI ZHEN  
SUPERVISORY PATENT EXAMINER

QC / QC  
January 24, 2007